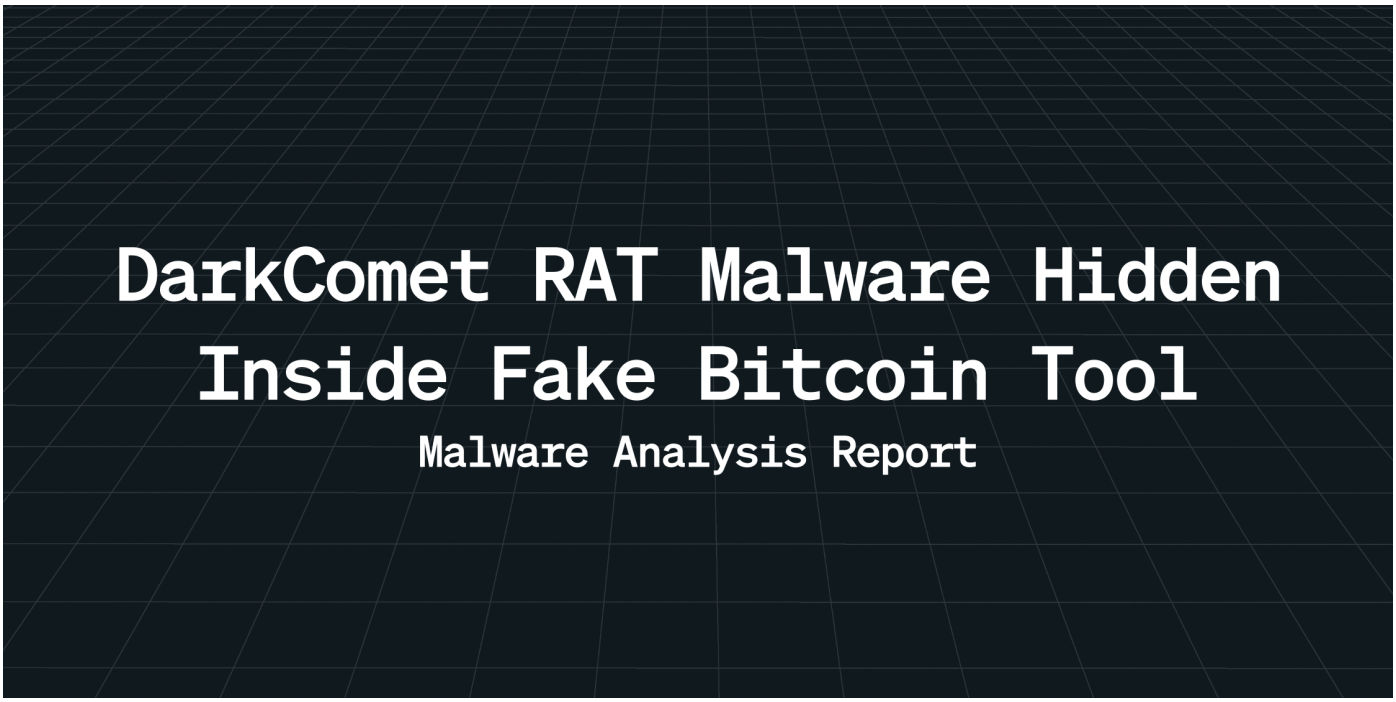
DarkComet RAT Malware Hidden Inside Fake Bitcoin Tool

pointwild.com/threat-intelligence/darkcomet-rat-malware-hidden-inside-fake-bitcoin-tool

Lat61 Threat Intelligence Team November 11, 2025



Introduction

The rise of cryptocurrency has not only transformed global finance but has also opened new opportunities for cybercriminals. Bitcoin, being the most recognized digital currency, has become a common lure in social engineering campaigns. Attackers frequently disguise their malware as Bitcoin wallets, mining software, or trading tools to trick unsuspecting users into executing them.

One such case is the reappearance of DarkComet RAT, a well-known remote access trojan that, despite being discontinued by its creator years ago, continues to circulate in underground forums and attack campaigns. DarkComet is notorious for its rich set of spying and control features, ranging from keystroke logging and file theft to webcam surveillance and remote desktop control.

The sample analyzed in this post masquerades as a Bitcoin-related application, appealing to individuals interested in cryptocurrency. Once installed, instead of delivering the promised functionality, it silently activates the full arsenal of “DarkComet RAT”. This highlights two key aspects of the modern threat landscape:

1. **Old malware never truly dies**: Once publicly leaked, families like DarkComet are repurposed indefinitely.
2. **Cryptocurrency remains a prime attack vector**: Lures involving Bitcoin or digital assets continue to be highly effective against both casual users and investors.

This blog takes a closer look at the technical behavior of this Bitcoin themed DarkComet variant, examining how it operates, what capabilities it delivers to the attacker, and why cryptocurrency enthusiasts should remain cautious when downloading wallet tools or trading utilities from unverified sources.

Initial Discovery

The malware sample was obtained in the form of a compressed RAR archive, a common tactic used by threat actors to evade detection and lower suspicion. By packaging malicious executables inside archive files, attackers aim to:

- **Bypass email and web filters** that may block direct executable attachments.
- **Reduce antivirus detection rates**, since the archive format can conceal malicious payloads until extraction.
- **Encourage user interaction**, as the victim must manually extract and run the file, which attackers often disguise as a legitimate tool.

Upon inspection, the RAR file contained a single executable masquerading as a **Bitcoin wallet or utility application**. The naming convention and iconography suggested that the attacker was targeting cryptocurrency enthusiasts, leveraging Bitcoin as the lure.

Key observations during triage included:

- **Archive Type**: RAR compressed file
- **Contained Payload**: Executable disguised as Bitcoin software
- **Likely Delivery Method**: Could have been distributed via phishing emails, malicious forums, or file-sharing platforms
- **Attacker Goal**: To trick users into extracting and launching the RAT under the belief that they were opening a cryptocurrency related program

The use of a RAR archive adds an additional layer of deception, as many users consider compressed files harmless and are more likely to extract them without thorough inspection. This approach aligns with common **malware delivery patterns**, where attackers combine social engineering (in this case, cryptocurrency interest) with technical evasion (RAR compression) to maximize infection success rates.

File Info of Compressed file

MD5: dbedd5e7481b84fc5fa82d21aa20106f

SHA-1: 87a2425098d257f4c0450a0cf56d0209963096d4

SHA256: 11bf1088d66bc3a63d16cc9334a05f214a25a47f39713400279e0823c97eb377

File Size: 255.23 KB

File Type: RAR

Upon decompression it decompresses to “94k BTC wallet.exe” which is UPX Packed

File Info of Decompressed file

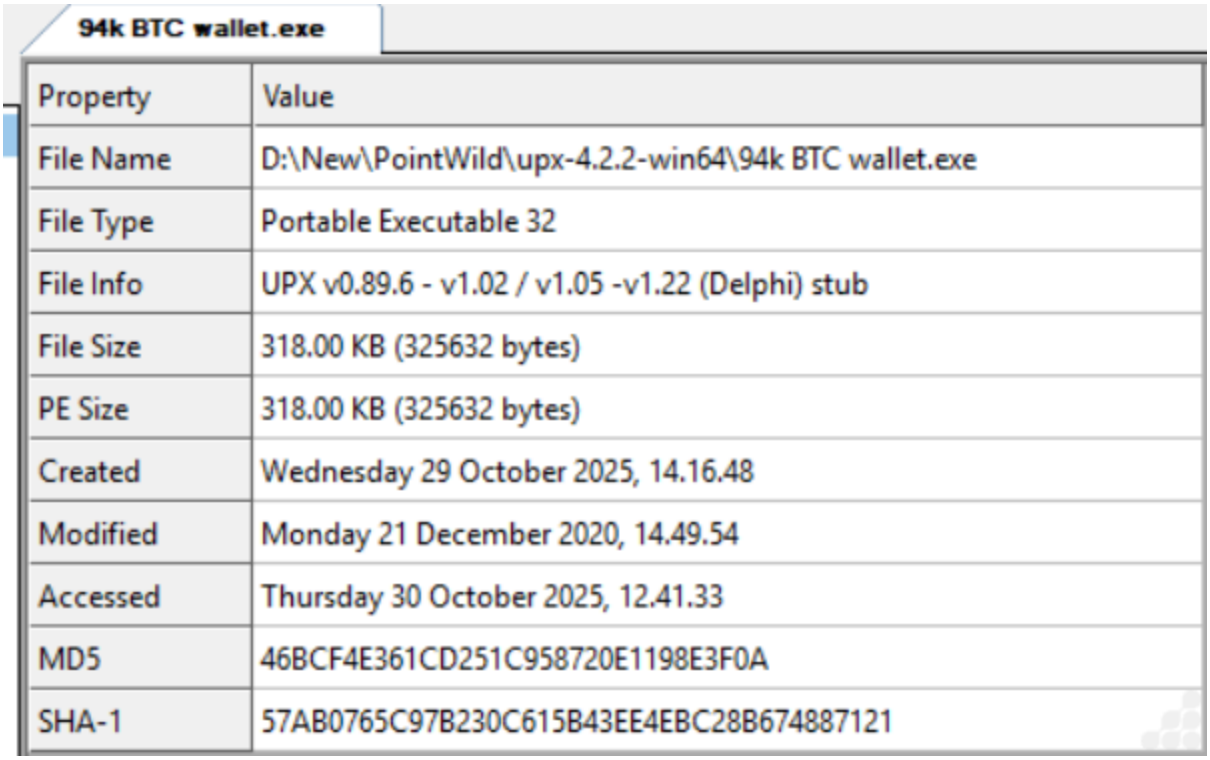


Figure 1: File Info image

File Name : 94k BTC wallet.exe

MD5 : 46bcf4e361cd251c958720e1198e3f0a

SHA-1 : 57ab0765c97b230c615b43ee4ebc28b674887121

SHA256 : 5b5c276ea74e1086e4835221da50865f872fe20cfc5ea9aa6a909a0b0b9a0554

File Size : 318.00 KB

File Type : Win32 EXE (Portable Executable 32)

PEiD packer : UPX v0.89.6 – v1.02 / v1.05 -v1.22 (Delphi) stub

Technical Analysis:

Unpacking a Suspicious Bitcoin Wallet Executable

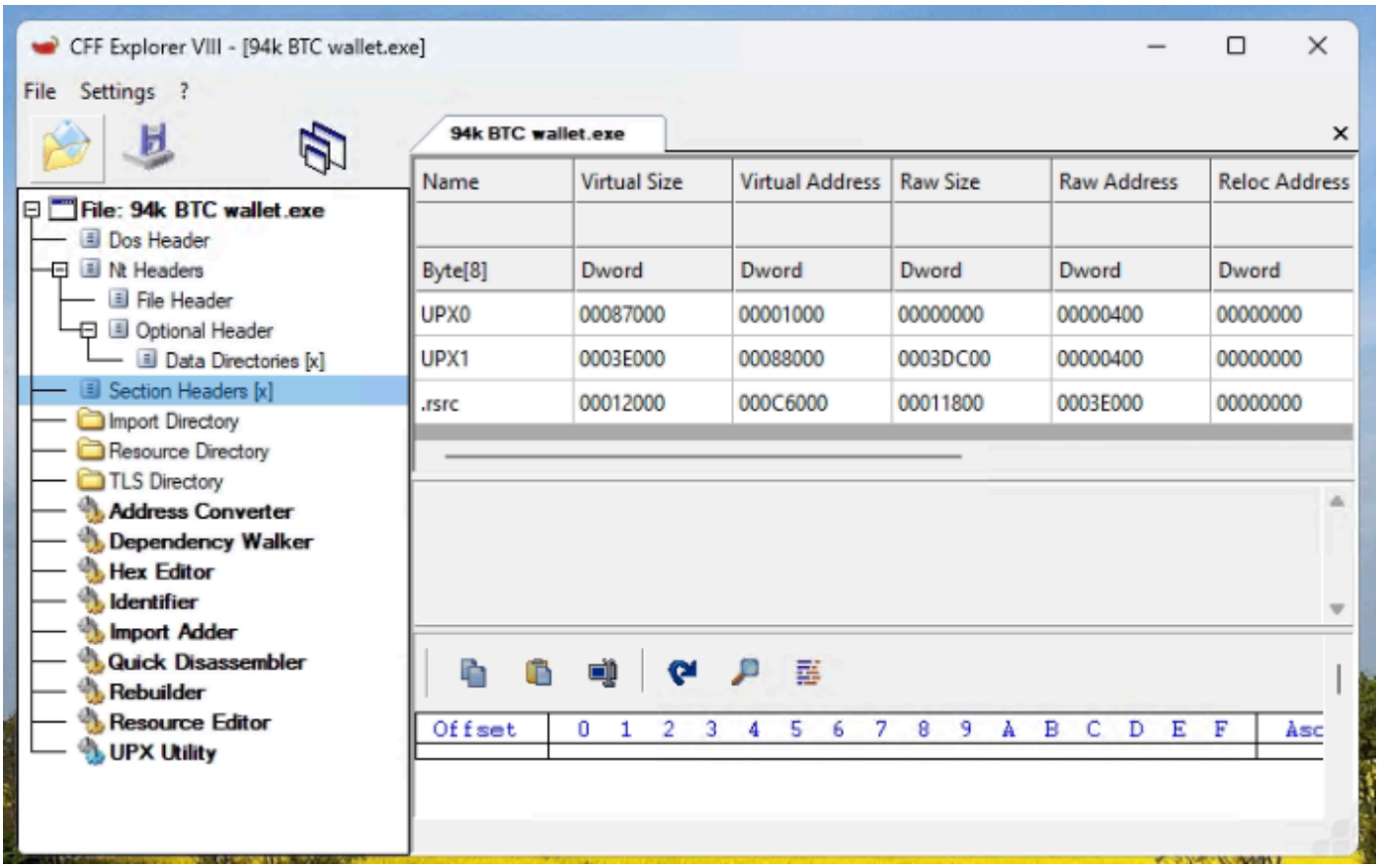


Figure 2: UPX Packed

During static analysis of the suspicious executable “94k BTC wallet.exe”, the first step was to check the PE (Portable Executable) structure using CFF Explorer. This revealed that the sample was packed with UPX (Ultimate Packer for Executables) a common method malware authors use to compress and obfuscate executables to evade detection.

Step 1 : Inspecting the Executable in CFF Explorer

When loading the file into CFF Explorer, the Section Headers tab showed the following key details:

Sections present:

- UPX0
- UPX1
- .rsrc

These are strong indicators that the file is packed with UPX. Typically, a clean Windows PE executable contains sections such as .text, .data, .rdata, and .rsrc. The presence of only UPX0 and UPX1 (instead of .text, .data, etc.) means that the code and data have been compressed into UPX containers.

Virtual & Raw Sizes:
The table shows that the UPX0 section has a large virtual size (00087000) but a very small raw size (00000000). This is a red flag: it indicates that the actual code is not directly available on disk but will be unpacked at runtime into memory.

Similarly, the UPX1 section (0003E000 raw size) contains the compressed payload that will be decompressed during execution.

Thus, static inspection confirms the binary is packed and not in its original form.

Step 2 : Unpacking the Executable with UPX Packer ?

```
Microsoft Windows [Version 10.0.26100.6899]
(c) Microsoft Corporation. All rights reserved.

D:\New\PointWild\upx-4.2.2-win64>upx.exe -d "94k BTC wallet.exe" -o Unpacked94kBTCwallet.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2024
UPX 4.2.2       Markus Oberhumer, Laszlo Molnar & John Reiser    Jan 3rd 2024

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
    742400 <-    325632   43.86%    win32/pe     Unpacked94kBTCwallet.exe

Unpacked 1 file.

D:\New\PointWild\upx-4.2.2-win64>
```

Figure 3: Successful unpacking of “94k BTC wallet.exe” with UPX 4.2.2

To analyze the actual malicious logic, the executable must be unpacked. The UPX tool provides a built-in decompression option. The following command was used:

upx.exe -d “94k BTC wallet.exe” -o Unpacked94kBTCwallet.exe

- -d → instructs UPX to decompress the binary.
- -o → specifies the output filename for the unpacked executable.

The unpacking process completed successfully, producing an unpacked file named Unpacked94kBTCwallet.exe.

- Original packed size: 325,632 bytes
- Unpacked size: 742,400 bytes
- Compression ratio: ~43.86%

This confirms that nearly half the original executable had been compressed, and the unpacked binary is now in a state suitable for deeper reverse engineering.

Step 3 : Why Does DarkComet Use UPX Packer ?

DarkComet RAT samples are often packed with UPX (Ultimate Packer for Executables) to make detection and analysis more difficult.
Malware authors behind DarkComet use UPX for the following reasons:

- **Evasion of Static Signatures:**
  By compressing and restructuring the executable layout, the DarkComet binary avoids static detections that rely on known byte patterns or file hashes.
- **Obfuscation of Code and Imports:**
  Packing hides the real API imports and code structure, which makes it harder for analysts and antivirus engines to understand the malware's behavior through static inspection.
- **Smaller Payload Size:**
  The packed executable is smaller in size, helping the attacker distribute it more easily via phishing emails or malicious downloaders.

During analysis, we observed that the DarkComet sample was UPX-packed, and after unpacking, our engine successfully detected and classified the payload as a **Backdoor.DarkComet**.
Although UPX is an open-source packer and easy to reverse with official tools, some DarkComet variants modify UPX headers or add extra encryption layers to slow down the analysis process.

File Info of Unpacked file

MD5 : d74ca6016bdde3df525d7c7651747336

SHA-1 : dc56a542e3db56f1c7132d3e99c960c09396cde3

SHA256 : 58c284e7bbeacb5e1f91596660d33d0407d138ae0be545f59027f8787da75eda

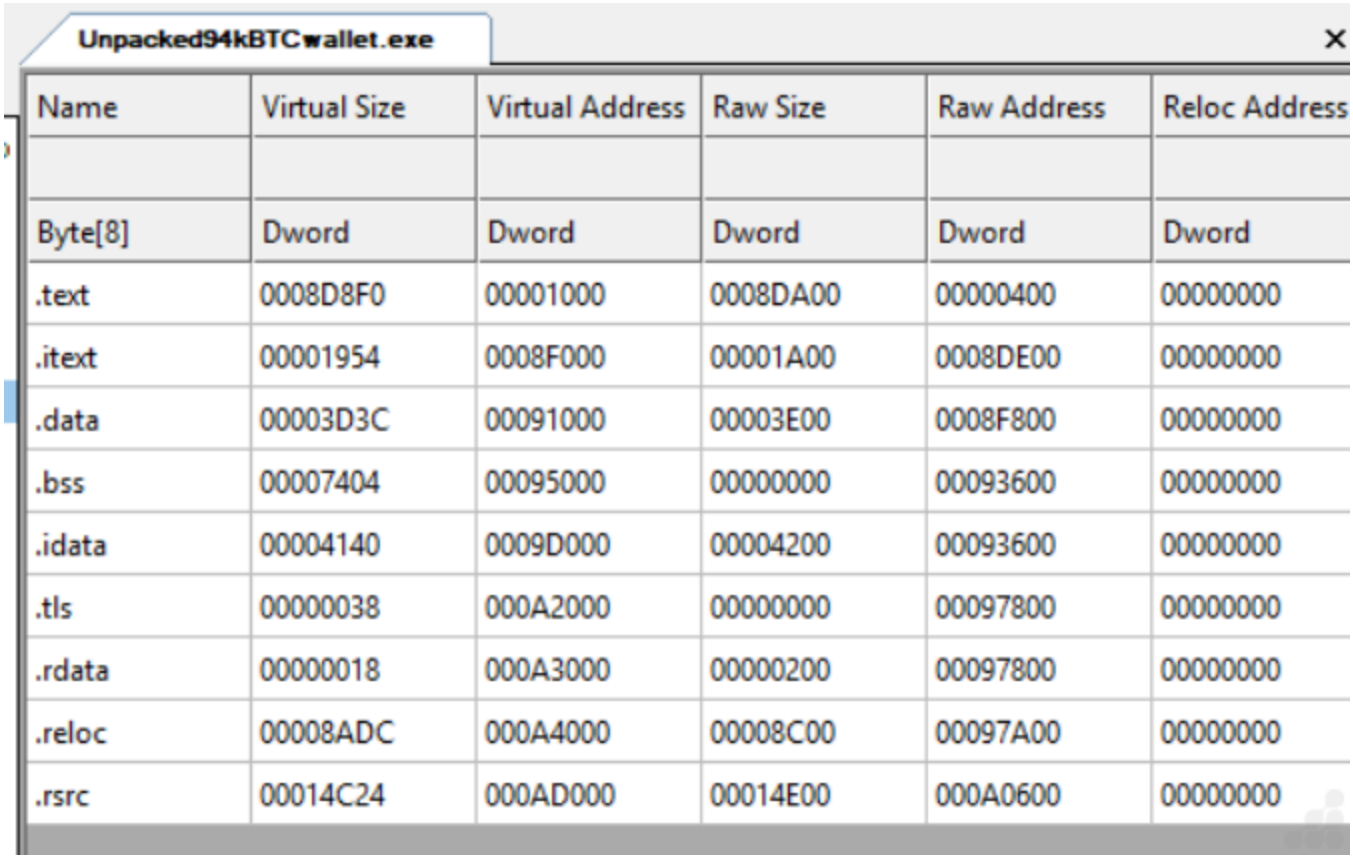
File Size : 725.00 KB

File Type : Win32 EXE (Portable Executable 32)

Compiler : Borland Delphi (2006) [Professional]

After unpacking the **UPX packed DarkComet sample (94kBTCwallet.exe)**, we examined the unpacked executable in a PE analysis tool to verify the restoration of original sections.

As shown below, the unpacked binary now displays multiple standard Portable Executable (PE) sections such as **.text**, **.data**, **.rdata**, **.idata**, and others which were previously compressed and hidden inside the UPX-packed version.



Unpacked94kBTCwallet.exe

Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address
Byte[8]	Dword	Dword	Dword	Dword	Dword
.text	0008D8F0	00001000	0008DA00	00000400	00000000
.itext	00001954	0008F000	00001A00	0008DE00	00000000
.data	00003D3C	00091000	00003E00	0008F800	00000000
.bss	00007404	00095000	00000000	00093600	00000000
.idata	00004140	0009D000	00004200	00093600	00000000
.tls	00000038	000A2000	00000000	00097800	00000000
.rdata	00000018	000A3000	00000200	00097800	00000000
.reloc	00008ADC	000A4000	00008C00	00097A00	00000000
.rsrc	00014C24	000AD000	00014E00	000A0600	00000000

Figure 4: Unpacked File Details

Persistence Mechanism

In Registry screenshot:

HKCU\Software\Microsoft\Windows\CurrentVersion\Run

└── Path -> C:\Users\admin\AppData\Roaming\MSDCSC\explorer.exe

Figure 5: Registry key

The binary copies itself asexplorer.exeunder%AppData%\Roaming\MSDCSC\ and creates a Run key for autostart.Ensures execution every system reboot.

Embedded DarkComet Configuration

Figure 6: Configuration of DarkComet in Memory

This reveals:

- Mutex:DC_MUTEX-ARULYYD→ ensures only one instance runs.
- C2 Server: kvejo991.ddns.net over port 1604.
- Install Path: MSDCSC\explorer.exe under user AppData.
- Persistence Flags: Install = 1, Offline keylogging enabled.

This is the hardcoded RAT config extracted post-UPX unpacking

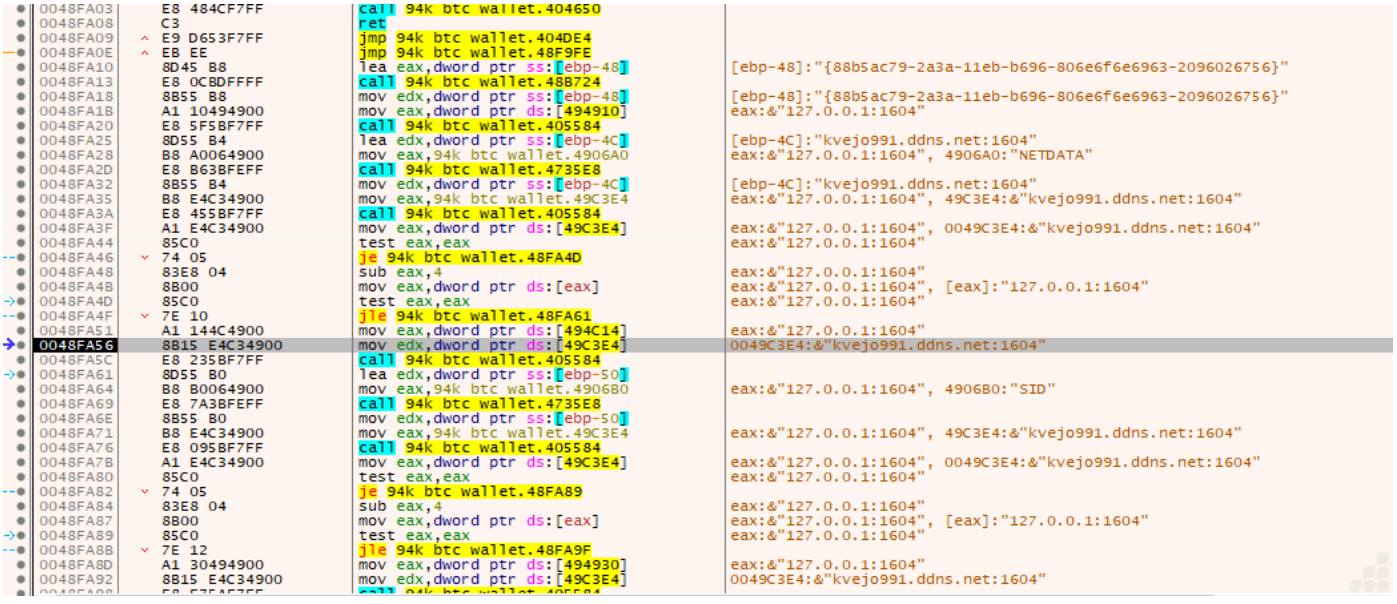


Figure 7: kvejo991.ddns.net:1604

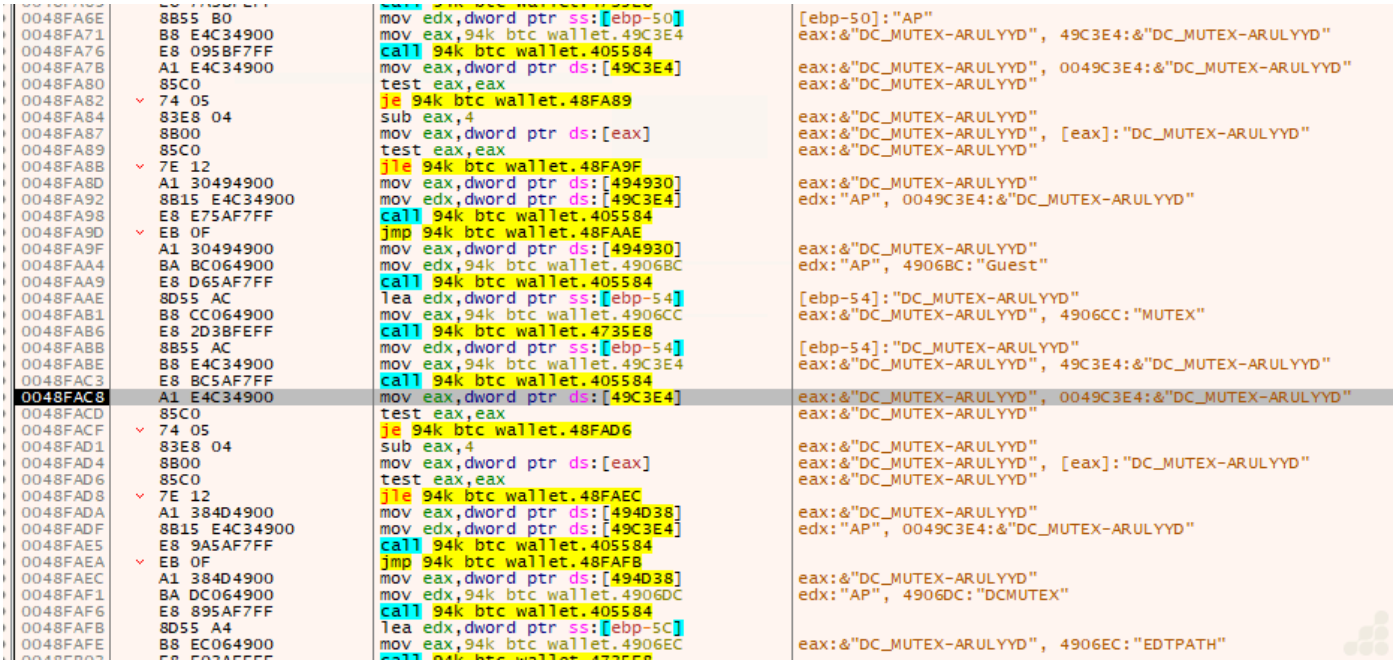


Figure 8: DC_MUTEX-ARULYYD

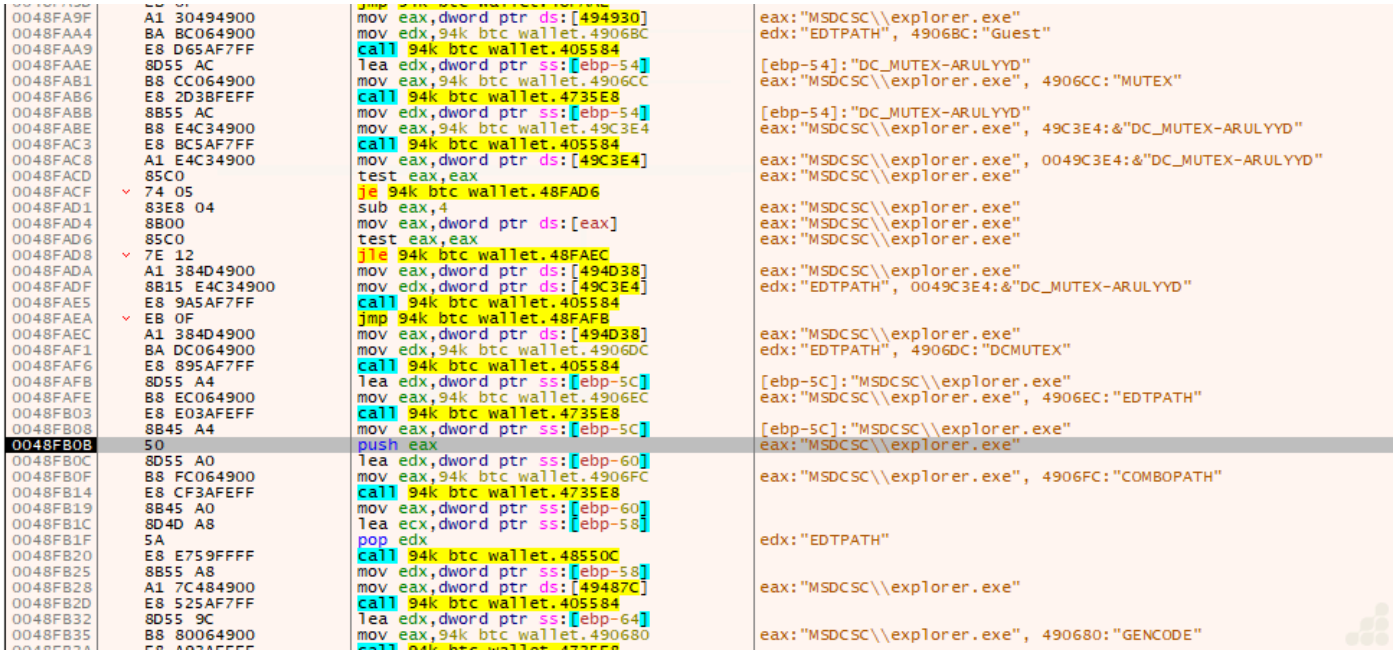


Figure 9: MSDSC\\explorer.exe

Keylogging Activity Captured Logs

During analysis, it was observed that the malware component performs keylogging activity, where it records the victim's keystrokes to capture sensitive information such as login credentials, chat messages, or banking details.

The captured keystrokes are then stored locally inside a folder named **dclogs**, which acts as the malware's data repository before the logs are either exfiltrated to the command-and-control (C2) server or retained for local collection by the attacker.

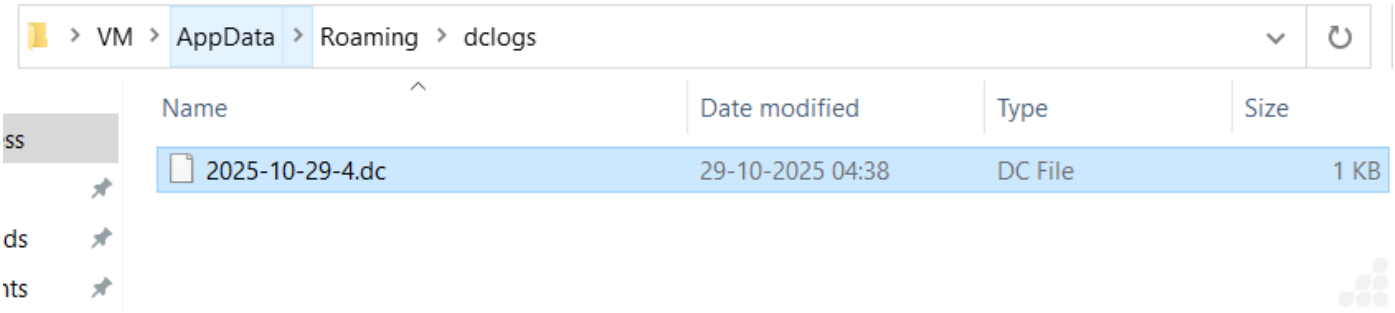


Figure 10: Keystroke activities captured in log file

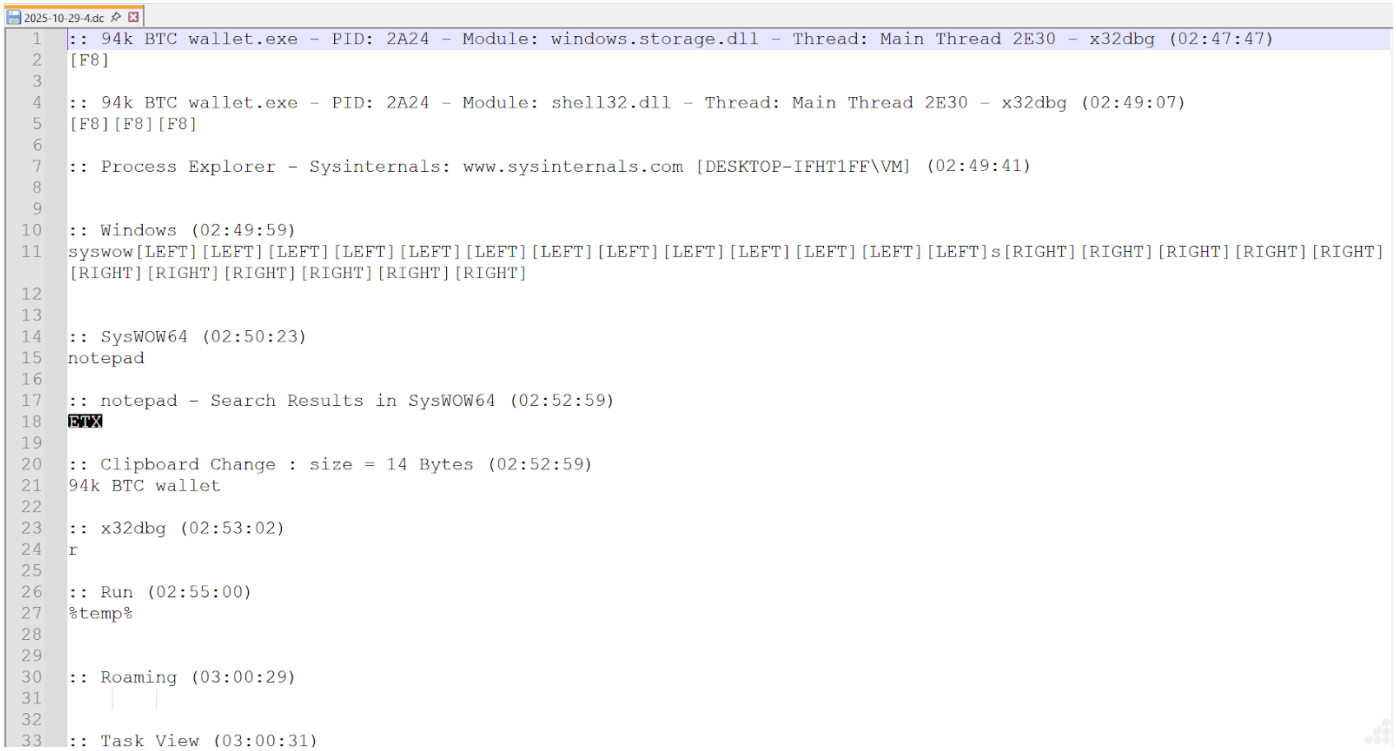


Figure 11: Keystroke logs

Process Behavior of “94k BTC wallet.exe”

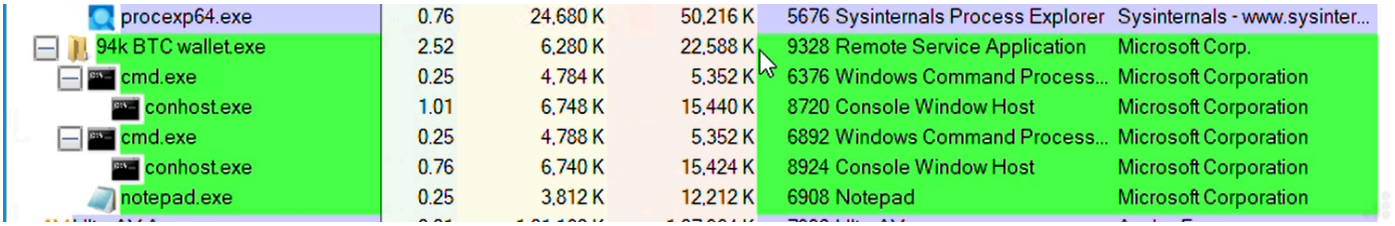


Figure 12: DarkComet initiates a stealthy injection chain, spawning multiple processes to cloak its malicious payload behind a deceptive interface.

Upon execution, the “94k BTC wallet.exe” spawns multiple cmd.exe and conhost.exe processes, as seen in the process hierarchy. These child processes indicate that the malware executes a series of internal commands to establish its runtime environment.

After the command shells are spawned, the malware further launches notepad.exe, which is a known decoy process behavior of DarkComet. The RAT injects its payload into Notepad's process space to perform actions such as keylogging, screen capture, and remote command execution, while keeping activity hidden under a legitimate Windows process.

Command-and-Control (C2) Communication

During network analysis, the unpacked DarkComet sample was observed attempting to establish a TCP connection to kvejo991.ddns.net on port 1604, which aligns with the default command-and-control (C2) port commonly used by the DarkComet RAT family. The connection logs showed multiple retransmissions, suggesting that the remote server was either offline or blocking incoming connections at the time of execution. Despite the failed connection, these repeated attempts clearly indicate active C2 beaconing behavior, confirming that the malware was trying to communicate with its operator to receive commands or exfiltrate data—behavior that is consistent with known DarkComet RAT activity patterns.

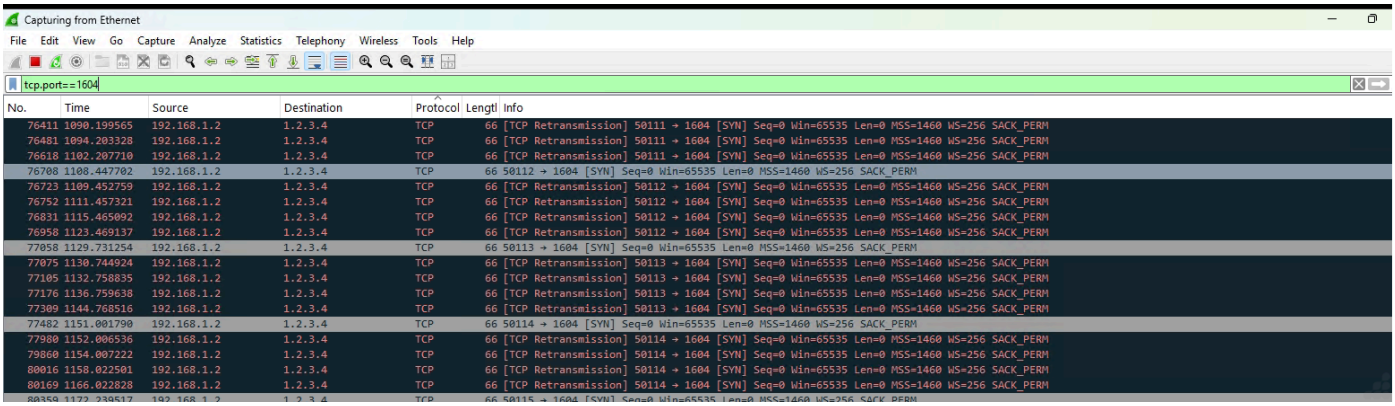


Figure 13: Network traffic on Port 1604

Indicators of Compromise (IOCs)

Category	Indicator
Archive File	11bf1088d66bc3a63d16cc9334a05f214a25a47f39713400279e0823c97eb377
Payload EXE	5b5c276ea74e1086e4835221da50865f872fe20cfc5ea9aa6a909a0b0b9a0554
Unpacked EXE	58c284e7bbeacb5e1f91596660d33d0407d138ae0be545f59027f8787da75eda
Install Path	C:\Users\<User>\AppData\Roaming\MSDCSC\explorer.exe
Registry Key	HKCU\Software\Microsoft\Windows\CurrentVersion\Run\explorer -> C:\Users\admin\AppData\Roaming\MSDCSC\explorer.exe
Mutex	DC_MUTEX-ARULYYD
C2 Domain	kvejo991.ddns.net
C2 Port	1604 (TCP)
Keystroke Capture Log file	2025-10-29-4.dc

MITRE ATT&CK Mapping

Tactic	Techniques	ID	Relevance in Sample
Initial Access	Spearphishing Attachment (Compressed Archive)	T1566.001	Delivered as a malicious RAR file attachment/download to lure victims with the Bitcoin tool theme.
Defense Evasion	Obfuscated/Compressed Binary – UPX Packing	T1027.002	The payload was packed with UPX to evade static detection.
Execution	User Execution	T1204	The victim manually extracts and runs the disguised Bitcoin application.
Persistence	Registry Run Keys / Startup Folder	T1547.001	DarkComet sets autostart entries to survive reboots.
Collection	Keylogging	T1056.001	Primary behavior observed: keystroke capture for credential and wallet theft
Command and Control	Application Layer Protocol	T1071.001	Establishes connection with C2 domain over TCP.
Command and Control	Exfiltration Over C2 Channel	T1041	Captured keystrokes and data are exfiltrated via the same C2 connection.

Removal

1. Reboot into Safe Mode with Networking.
2. Useupdated UltraAV.
3. UltraAV unpacked a UPX-packed executable into the TempData folder and detected the malicious payload with following name:

   Backdoor.DarkComet

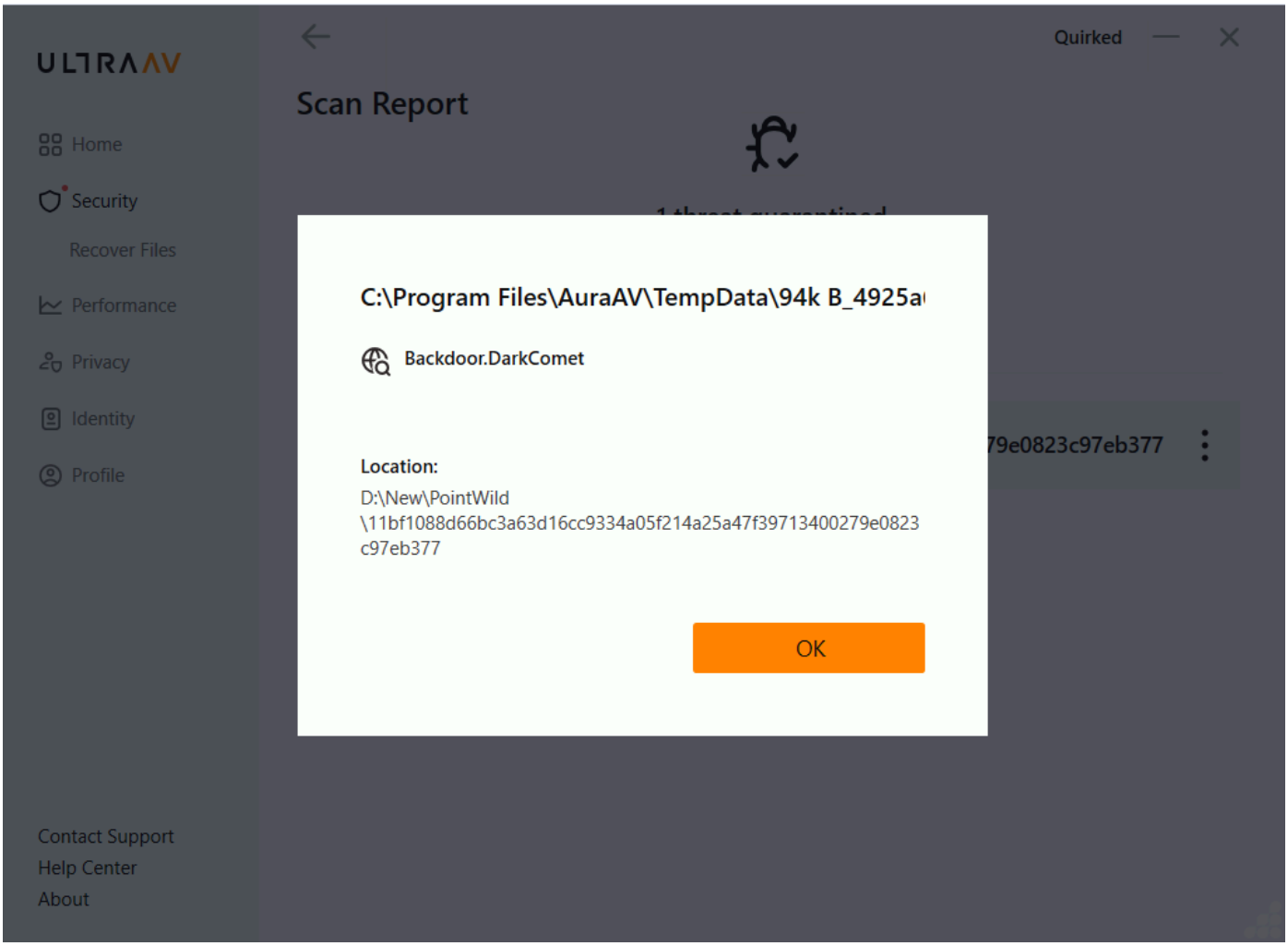


Figure 14: Ultra AV Detection

Conclusion

The analysis of this Bitcoin themed DarkComet RAT sample demonstrates how old malware families continue to find new life through modern lures. By hiding inside a file packaged as a cryptocurrency utility, the attacker leveraged the ongoing hype around Bitcoin to trick users into executing a well established remote access trojan. Although DarkComet is not a new threat, its feature set remains dangerous: keylogging, credential theft, file manipulation, surveillance, and persistence techniques are still effective against unsuspecting victims. Combined with the lure of cryptocurrency applications, this makes the malware especially impactful, since compromised systems may lead directly to stolen wallet credentials and financial losses.